Cracking the Universal Product Code

by Count Nibble

Everyone encounters the UPC nowadays. You know, it's that set of black bars you see on virtually every product whenever you go to the grocery store, to buy a book or a magazine, or even to buy software (assuming that you do, indeed, BUY your software). Have you ever though of what fun you could have by altering that little set of black bars? If you were lucky enough, you might be able to slip a box of industrial size laundry detergent by that dizzy 16-year-old girl at the Safeway and have the computer charge you the price of a pack of Juicy Fruit, or some other such mischief. Well, to help you in your explorations of How To Screw Over Others In This Grand Old Computerized World of Ours, I proudly present HOW TO CRACK TO UPC CODE. Use the information contained herein as you will. You will need the file UPC.PIC, hopefully available from the same place you found this file. And so, let's begin:

When the lady at the corner market runs the package over the scanner (or whatever it is they do in your area), the computerized cash register reads the UPC code as a string of binary digits. First it finds the "frame bars" - a sequence of "101" (see A on picture). There are three sets of frame bars on any given code...one on either side, and one in the center. These do nothing but set off the rest of the data, and are the same on any UPC code. Next is the "number system character" digit, which is encoded in leftside code (see later). This digit tells the computer what type of merchandise is being purchased. The digits and their meanings are:

0 - Ordinary grocery items. Bread, magazines, soup, etc.
2 - Variable-weight items. Meats, fruits & veggies, etc.
3 - Health items. Aspirin, bandaids, tampons, etc.
5 - Cents-off coupon. (Not sure how this works).

The next cluster of digits is the manufacturer number, again stored in leftside code. THere are five digits here all the time. Some numbers include 51000 for Campbell's Soup, 14024 for Ziff-Davis publishing (Creative Computing, A...), and 51051 for Infocom. The next five digits (after the frame bars) are the product/size id number. The number for "The Hitchhiker's Guide to the Galaxy" from Infocom is 01191. These digits are stored in rightside code. Finally there is the checksum, in rightside, which will be discussed later.

Now, why are there two types of codes, leftside and rightside? That's so the person at the checkout counter can slide the thing by the scanner any way she pleases. By having different codings for either side the computer can tell the right value no matter how the digits are read in. Here are the codes for the digits 0 through 9:

Digit	Leftside code	Rightside code
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

The more observant among you may have noticed that Rightside code is nothing more than logical-NOTed Leftside code, i.e., a 0 in Leftside is a 1 in Right-side, and vice versa. Later on we will discuss another type called Reversed Rightside, in which the binary values in Rightside are reversed, meaning that 1110100 (9) in Rightside would be 0010111 in Reversed Rightside. RR is used only when there is an extra set of codes off to the right of the main code bars, as with books and magazines.

Now we see the hard part: how the checksum digit is encoded. Let's try working out the checksum for "Hitchhiker's Guide".

First, notice the Number System Character. Software is considered a Grocery Item by UPC, so the NSC is 0 (zero). Next, Infocom's Manufacturer's Number is 51051, and the game's id number is 01191. Good enough. Set together, these numbers look like this:

```
0 51051 01191
```

Now, take the digits of the code and write them on alternate lines, odd on one line, even below, giving this:

```
0 1 5 0 1 1
 5 0 1 1 9
```

Now add each set of numbers:

```
0+1+5+0+1+1 = 8
 5+0+1+1+9 = 16
```

Multiply the first number (the ones created by adding the first, third, etc digits) by three:

```
8x3 = 24
```

And add that to the result of the other number (second, fourth, etc digits added together):

```
24+16=40
```

Subtract this from the next higher or equal multiple of 10 (40 in this case)

```
40-40=0
```

And the remainder, here 0 (zero), is the checksum digit.

Now, what if there's a set of other bars off to the side? These are encoded in another format which uses Reversed Rightside (as described above) instead of standard Rightside. For books, the sequence is as follows:

```
Five digits
Starts with 1011
If (first digit is even) then
        sequence is L-RR-L-L-RR
else
        sequence is RR-L-L-RR-L
each digit is separated with 01
```

Therefore, the sequence for 29656 is:

```
1011 0010011 01 0010111 01 0101111 01 0110001 01 0000101
       2L         9RR        6L         5L         6RR
```

and the sequence for 14032 is:

```
1011 0110011 01 0100011 01 0001101 01 0100001 01 0010011
       1RR        4L         0L         3RR        2L
```

Naturally, all these bars are run together. There is no checksum.

For magazines, the sequence is even more complex. There are two digits in each bar, and the numbers usually run from 1-12, signifying the month. The first digits are encoded thusly:

```
L if the digit is 1,4,5,8 or 9 and
RR if the digit is 2,3,6,7 or 0.
```

The second digit is coded in L if it is even, and RR if it is odd. Therefore,

06 codes as:

        1011 0100111 01 0101111

and 11 codes as:

        1011 0110011 01 0110011

No checksum here, either, and the fields are again separated by 01.

Well, that about does it for this explanation of how to crack the UPC codes. Use this information as you will, and forward any question to THE SPACE BAR, xxx-xxx-xxxx, pw:BANZAI. Enjoy!

        - Count Nibble -

The PIRATES HOLLOW   xxx-xxx-xxxx ;(